Systems Engineering Advancement Research Initiative

SEAri Working Paper Series

Title: A Prescriptive Semantic Basis for System Lifecycle Properties

Author: A.M. Ross, J.C. Beesemyer, and D.H. Rhodes

Paper Number: **WP-2011-2-2**

Revision Date: December 16, 2012

The content in this paper is in pre-published, draft form. It has not been peer-reviewed and should be used for informational purposes only. The authors warn that the paper may contain typos, misquotes, incomplete, and possibly incorrect content. It is intended that the paper will be revised and elevated to "published" status, at which point the quality caveat will be lifted.

A Prescriptive Semantic Basis for System Lifecycle Properties

Adam M. Ross, J. Clark Beesemyer, and Donna H. Rhodes

Systems Engineering Advancement Research Initiative (SEAri)
Massachusetts Institute of Technology

E38-575, 77 Massachusetts Ave
Cambridge, MA 02139

.

Abstract. The concept of incorporating "ilities" into systems seems a self-evidently good idea, as expressed by both written and spoken positions by technical and political leaders. Indeed, incorporating lifecycle properties such as flexibility, adaptability, and recently affordability, into systems is touted as the solution to modern day's ever increasing complexity, schedule and budget pressures, and need for finding sustainable solutions. While expressing desires for ilities seems straightforward, tracing these desires to verifiable system instantiations remains ambiguous at best. This paper describes the semantic challenge underlying the concept of a coherent set of system properties, a sampling of various efforts to ascribe meaning to particular ilities, and proposes a prescriptive semantic basis for specifying a set of ilities, while avoiding the assertion of new definitions. The intention for this first pass prescriptive semantic basis is to begin a structured approach for exploring the existence of one or more semantic fields, which together form a coherent semantic framework for tracing desired ilities into verifiable system requirements and specifications. Preliminary results indicate that at least three semantic fields exist within the larger set of system lifecycle properties including change-type, architecture-type, and new ability-type ilities.

Motivation

These days, it is common to hear terms such as "flexible" or "adaptable" or "robust" in the parlance of politicians and technical leaders in the U.S. and other nations.

When terms such as these are used in speech or text, the precise meaning can sometimes be kept intentionally vague in order to convey generalized impression of the intent of the speaker. In policy speeches, generalized impressions are more likely to garner support through various interpretations by the listeners. When it comes to implementation however, precise meaning becomes more important in order to validate that the mandates have been successfully carried out.

> *"Rep. John Mica called on the agency to "reform" and "become...a thinking, risk-based, **flexible** agency that analyzes risks, sets security standards and audits security performance."*
>
> *"Defense Secretary Panetta: "The US joint force will be smaller and it will be leaner. But it will be more **agile**, more **flexible**, ready to deploy quickly, innovative and technologically advanced."*
>
> *… "the Defense Department and the Office of the Director of National Intelligence pledged to foster an industrial base that is **'robust**, competitive, **flexible**, healthy, and delivers **reliable** space capabilities on time and on budget.'"*
>
> Quotes from AIAA Daily Launch, 20 Jul 2011 – 13 Feb 2012

In order to better understand these "ilities" enough to ensure that systems predictably display these properties, this paper will propose a semantic-approach to frame the problem of disambiguating possible semantic field structure among the so-called "ilities." Semantics is the scientific study of "meaning" and is a promising area for clarifying the "ilities." In this context, it is important to recognize that meaning arises from the interplay of "use" (i.e. in speech) and "prescription" (i.e. definitions and dictionaries). Ideally meaning is universal and well defined, with direct correspondence between the prescriptive meaning and "use" meaning of terms. In practice, the correspondence changes over time with meaning in one sense impacting meaning in the other sense. A particularly relevant concept from semantics is that of the ***semantic field***. A semantic field is *a group of words with related meanings, for example, kinship terms or color terms* (Akmajian et al. 2001, p. 239-240).

Semantic challenges for ilities

One of the fundamental challenges for developing a clearer understanding of the semantics of "ilities" is the current ambiguity in these terms. Many of these terms are used colloquially and therefore inherit informal meaning. Additionally, the terms, as currently used, display polysemy and synonymy. Polysemy is "the property of [a term] having multiple meanings that are semantically related" (Akmajian et al. 2001, p. 585). An example of polysemy is two different, but related meanings for flexibility: "able to be changed" and "able to satisfy multiple needs"[i]. In contrast to polysemy, synonymy is "the property of multiple terms having similar meaning." An example of synonymy is the interchangeable use of flexibility (able to be changed) and changeability (able to be changed or change itself).

One of the reasons for this ambiguity in the technical usage of ilities is that typically ilities are mostly considered one at a time in the literature. As an example, consider flexibility in Saleh, Mark, and Jordan (2009), Nilchiani (2005), and de Neufville and Scholtes (2011). Some work has been done on sets of ilities, such as for changeability in Fricke and Schulz (2005), and Ross, Rhodes, and Hastings (2008).

Addressing sets of ilities

If the challenge of understanding the semantic relationships amongst the ilities can be addressed by looking at sets of ilities, how should the members of the sets be selected? Many approaches have been attempted in the past: academic expert opinion, practitioner interviews, crowd-sourced, informed assertions, standards, descriptive surveys, and prescriptive assertions. These are briefly described below.

Academic expert opinion

In 2001, a committee of 11 MIT faculty developed a working paper called "ESD Terms and Definitions" in order to lay out the key concepts for the emerging field of Engineering Systems (ESD 2001). These 14 ilities are listed in Table 1. Ilities, as defined by this paper are: "requirements of systems, such as flexibility or maintainability, often ending in 'ility;'

[i] For a broad discussion of multiple related meanings for flexibility see Saleh, Mark, and Jordan (2009).

requirements of systems that are not necessarily part of the fundamental set of functions or constraints" (ESD 2001, p. 4). On inspection of the list and the faculty listed as co-authors, the list appears to be reflective of the research interests of the committee members.

Table 1. ESD Proposed ilities

ID Number	Name	ID Number	Name
P1	Flexibility	P8	Safety
P2	Agility	P9	Durability
P3	Robustness	P10	Sustainability
P4	Fail-safe	P11	Quality
P5	Adaptability	P12	Reliability
P6	Scalability	P13	Repairability
P7	Modularity	P14	Maintainability

Practitioner interviews

An alternative approach to determining important ilities is to ask practitioners about which ilities are well understood and which are important for their systems. Ross (2006) describes an exploratory set of surveys and interviews conducted with a number of Federally Funded Research and Development Centers (FFRDCs) and Prime contractors in the aerospace sector. These two interviewee types were presented with the list of MIT ESD-proposed ilities, along with manufacturability, testability, and operability, which were ilities that arose as important during initial test interviews among practitioners.

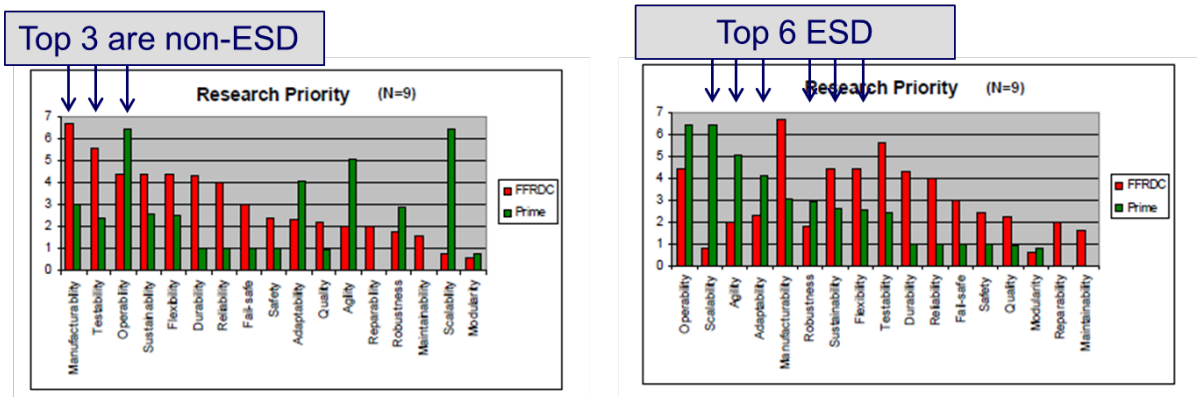


Figure 1. Reported ility interests sorted by research priority for FFRDC (left) and Prime (right)

Figure 1 illustrates the results from these surveys, showing little agreement between the FFRDCs and Primes, except that manufacturability, testability, and operability were actually quite important. According to the Primes (which are more like practitioners than the FFRDCs), the top six research priorities for (ESD-named) ilities are:

1. Scalability
2. Agility
3. Adaptability
4. Robustness
5. Sustainability
6. Flexibility

Subsequent interviews revealed that in spite of the talk by senior leaders and politicians of the desire for ilities, "'ilities' are not part of the lexicon of even the most successful project managers," so somewhere in the technical communication chain, the desire for ilities is not being translated into actionable concepts at the project level (Ross 2006, p. 37).

Crowd-sourced

Another alternative approach to identifying ilities is to use a crowd-sourced method, with one notable example being Wikipedia. The Wikipedia page for "ilities" defines the concept as follows: "within systems engineering, quality attributes are non-functional requirements used to evaluate the performance of a system. These are sometimes named "ilities" after the suffix that many of the words share" (Wikipedia 2012). The number of ilities reported on this webpage has grown from 61 (on 24 April 2008) to 71 (on 3 March 2010) to 79 (on 8 Feb 2011) to 81 (on 13 Feb 2012). The current list is shown in Figure 2 below. Of note is that changeability, manufacturability, quality, reconfigurability, and versatility are missing from the list.

accessibility	deployability	modularity	supportability)
accountability	discoverability	operability	securability
accuracy	distributability	orthogonality	simplicity
adaptability	durability	portability	stability
administrability	effectiveness	precision	standards compliance
affordability	efficiency	predictability	survivability
agility	evolvability	producibility	sustainability
auditability	extensibility	provability	tailorability
autonomy	failure transparency	recoverability	testability
availability	fault-tolerance	relevance	timeliness
credibility	fidelity	reliability	traceability
process capabilities	flexibility	repeatability	ubiquity
compatibility	inspectability	reproducibility	understandability
composability	installability	resilience	upgradability
configurability	Integrity	responsiveness	usability
correctness	interchangeability	reusability	
customizability	interoperability	robustness	
debugability	learnability	safety	
degradability	maintainability	scalability	
determinability	manageability	seamlessness	
demonstrability	mobility	self-sustainability	
dependability	modifiability	serviceability (a.k.a.	

Figure 2. Wikipedia list of ilities from 13 February 2012 (Wikipedia 2012)

Informed assertions

Illustrative of many uses of ilities in practice and literature is the selection of ilities based on brainstorming from experience and literature (Cotton et al. 2009). In one paper a large number of ilities were considered (more than 120) and down-selected through the efforts of the authors based on introspection and iteration in answering the following questions: "what are the overall objectives? What values are essential to ensuring effective [system] protection? What values are essential to architectures?" This study recognized that "no standard list of applicable "ilities" exist…" and "almost any attribute may be created by

adding '-ility' to the end of the word…" (Cotton et al. 2009, p. 3). The resulting sets of ilities were assigned to "architecture-focused properties" and "system-focused properties." These ilities are illustrated in Figure 3 below.

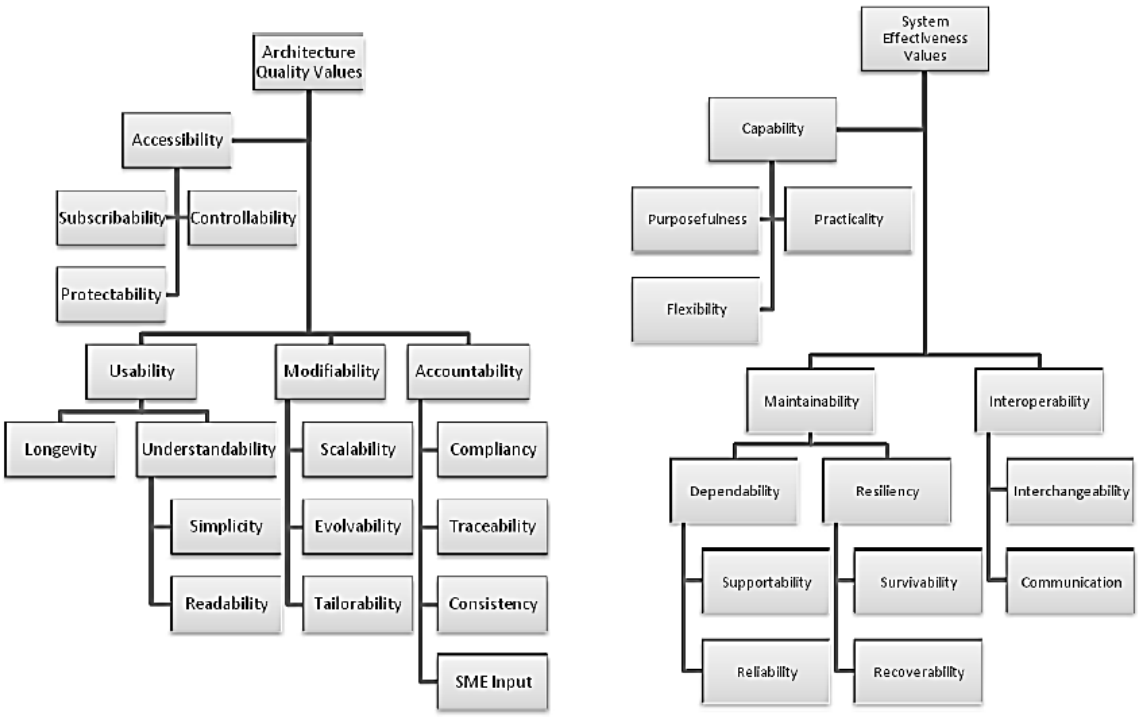


Figure 3. Ilities hierarchies for architecture quality (left) and system effectiveness (right) (figure from Cotton et al. 2009)

Standards for relationships

The ISO/IEC 9126-1 standard describes a set of software quality characteristics and sub-characteristics within a quality model. The model has six groups of high level ilities, with several ilities considered as lower level ilities within each group. The groups are: functionality, reliability, usability, efficiency, maintainability, and portability (ISO/IEC 2001). These ilities are described by Engel and Browning (2008) as subcomponents of adaptability, which is asserted as the over-arching ility (illustrated in Figure 4 below).

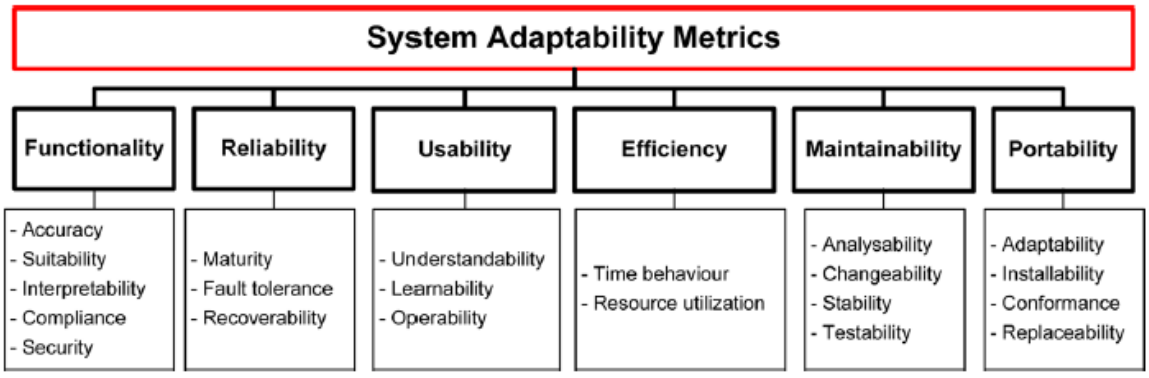


Figure 4. ISO/IEC 126-1 standard components (figure from Engel and Browning (2008)

While the standard describes various ilities within the context of software engineering, the content of this standard mixes "ilities" and software-specific quality measures. Additional

standards, as well as common groups of ilities exist, for example RAMS (reliability, availability, maintainability, and safety) and FURPS (functionality, usability, reliability, performance, and supportability) (Wikipedia 2012).

Descriptive surveys

A descriptive approach to determining ilities is described in de Weck, Magee and Roos (2011). Chapter 4: "Lifecycle Properties of Engineering Systems," describes various ilities and how their citation frequencies have changed over time. In particular, the frequency of ilities mentioned in journal articles and Google hits are compared to show the relative descriptive occurrence of various ilities in these written media. In an effort to understand the temporal relevancy within this data, an alternative representation shows the frequency of various ilities mentioned as a function of time. This shows the rise in mentioning of various ilities across distinct time periods (i.e. "epochs" of predominant system types), including the more recent rise of interoperability, sustainability, modularity, and testability.

While illustrative of the fact that not all ilities are mentioned with similar frequency, the work begs the question of whether frequency correlates with importance or relevance of the ilities relative to one another or in an absolute sense. The challenge is that these results do not capture relationship amongst ilities or the reasons behind the frequencies.

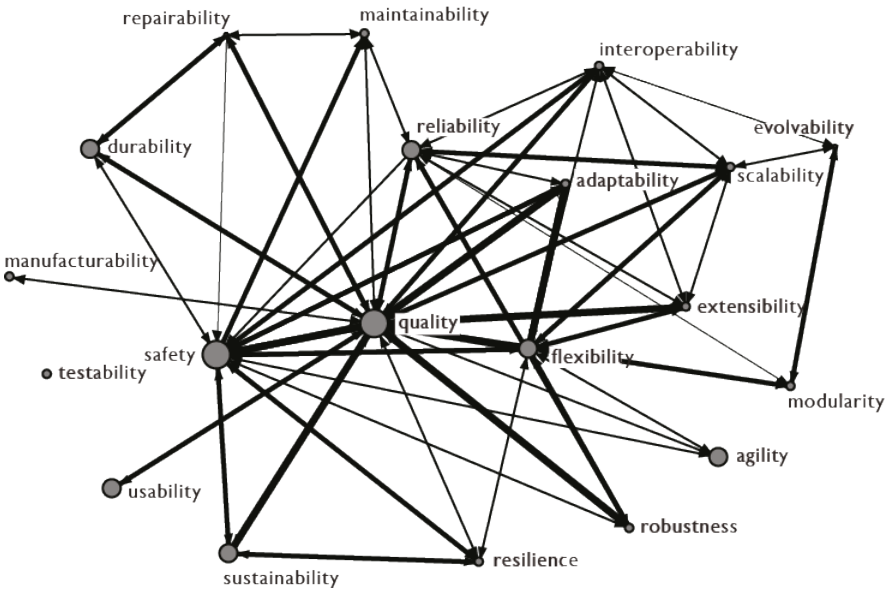


Figure 5. Ility co-occurrence in the literature, with implied dependence (figure from de Weck, Roos and Magee (2011), chapter 4)

In a first step to answering this question, a related descriptive study is also described in de Weck, Magee, and Roos (2011), which looks at the co-occurrence of ility terms in the literature, with implied dependence amongst the terms. A directed graph depiction (Figure 5) tempts the reader to begin to read causal relationships into the links between ilities, but the existence of "co-occurrence" cannot describe the nature of the link, which could include complements, substitutions, or other tradeoffs between pairs of ilities. These descriptive surveys are also described in de Weck, Ross, and Rhodes (2012). The descriptive results are a good first step for proposing deeper inquiry into the nature of the relationship amongst ilities, but are sensitive to the particular list of ilities investigated, and does not indicate whether polysemy or synonymy confounds the results (i.e. the distinctiveness and consistency of the semantic content of a given term).

Prescriptive assertions

As a complementary approach to discovering relationships amongst ilities, prescriptive assertions can be based upon theory or experience, making conceptual leaps in proposing how ilities should relate to one another. Prescriptive semantic approaches, such as definitions in dictionaries, may not agree completely with one another and are dependent on particular assumptions. The strength of prescriptive approaches is the intended self-consistency within a particular approach, as well as transparency in terms of relationships derived from explicit assumptions. Weaknesses of prescriptive approaches include the appropriateness of assumptions and the universality of the results. As an example, one may look at several sources for definitions of terms, such as Oxford English Dictionary[2] or Merriam-Webster[3] dictionary. The definitions for particular terms will not be in exact agreement, but will tend to be similar. The similarity is based partly on the approach for generating these definitions, which is embedded in the linguistic research of the staff responsible for the definitions. At some point, casual dictionary users came to assume that the proposed definitions in the dictionaries are "correct" and therefore seek to adhere to these prescriptive meanings in their own usage of terms. More sophisticated users may be more familiar with the techniques used to generate the definitions within particular dictionaries (for example the Oxford English Dictionary explicitly traces the historical usage and evolution of meaning for terms, enabling users to trace assumptions regarding semantic stability).

In the domain of ility-related prescriptive approaches, at least two types of work have been proposed: experience-informed approaches and theory-informed approaches. Both of these approaches utilize induction and deduction, with the former rooted more in induction and the latter more in deduction.

Change-type ilities

One of the key papers asserting a relationship amongst a set of ilities is the one by Fricke and Schulz (2005) that uses the concept of "changeability" as a higher order ility that encompasses four key ilities: adaptability, robustness, flexibility, and agility, illustrated in Figure 6.

[2] Oxford University Press, "the definitive record of the English language," available online at: http://www.oed.com.

[3] An Encyclopedia Britannica Company, available online at: http://www.merriam-webster.com.

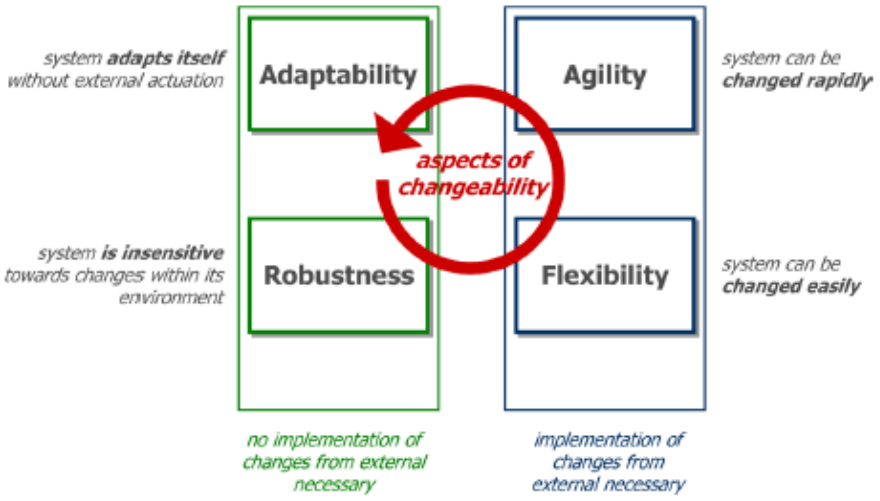


Figure 6. Changeability as four ilities (figure from Fricke and Schulz 2005)

In this work, a number of other ilities are mentioned in the context of "architecture principles" for achieving the changeability-related ilities. These include: simplicity, independence, modularity, integrability, autonomy, scalability, non-hierarchy, decentralization, and redundancy. Implicit in the paper is evolvability. The prescriptive nature of the framework for relating the ilities into change-type and architecture-type is based on the authors' research and experiences in German product development (e.g. BMW).

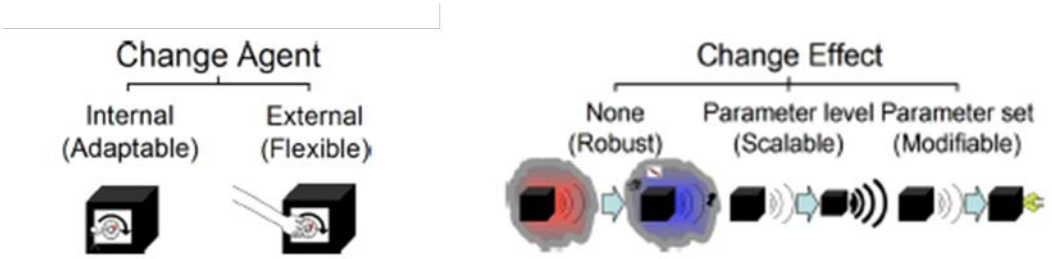


Figure 7. Change agent and effect dichotomy for changeability (figure from Ross, Rhodes and Hastings 2008)

Another changeability-related work is that of Ross (2006) and Ross, Rhodes, and Hastings (2008), which similar to Fricke and Schulz, asserts "changeability" as an overarching ility, with five underlying related ilities along two relationship dimensions: adaptability, flexibility (change agent), and robustness, scalability, and modifiability (change effect) (Figure 7). The concepts of change agent, change effect, and change mechanism are introduced as a means to generalize the concept of changeability and to provide a simple basis for deriving the other five ilities (i.e. adaptable scalability is an internal change agent instigating a change in the level of a system parameter). The "adaptable" vs. "flexible" ility label is dependent on whether the change agent is internal or external to the system boundary. This work introduces the concept that changeability is relative to one or more parameters of a system. In this way a system could display all five ilities. Additionally, this work introduced the concept of a verifiable changeability statement as a first step towards concretizing the concepts of these ilities into actionable project engineering and requirements (Figure 8).

The system shall be	_____	_____	in	_____	for less than	_____.
	(change agent type) flexibly or adaptably	(change effects) scalable, modifiable		(system parameter) with range		(resources)

Figure 8. Template for verifiable changeability requirement (from Ross, Rhodes, and Hastings 2008)

Means-ends hierarchy from prescriptive definitions

Recently, an initial exploratory study sought to uncover potential means-ends hierarchical relationships amongst a set of ilities (de Weck, Ross, and Rhodes 2012). In this study, four groups of graduate students constructed means-ends hierarchies from a given set of ilities. Each of the four groups independently derived distinct "hierarchies." In addition to connecting ilities with "means-ends" links, each of the four groups independently proposed a second grouping criterion called "level" or "depth" and structured their group hierarchies to display this quality. Figure 9 below illustrates the aggregate of the four hierarchies, with solid lines indicating 3 or 4 groups in agreement of means-ends relationship between two ilities, and dashed lines indicating 2 groups in agreement. The vertical placement of each ility corresponds to the median level assigned to that ility across the four groups. The lack of consensus and emergent "depth" criterion suggests that more than "means-ends" relationships exist amongst the ilities.

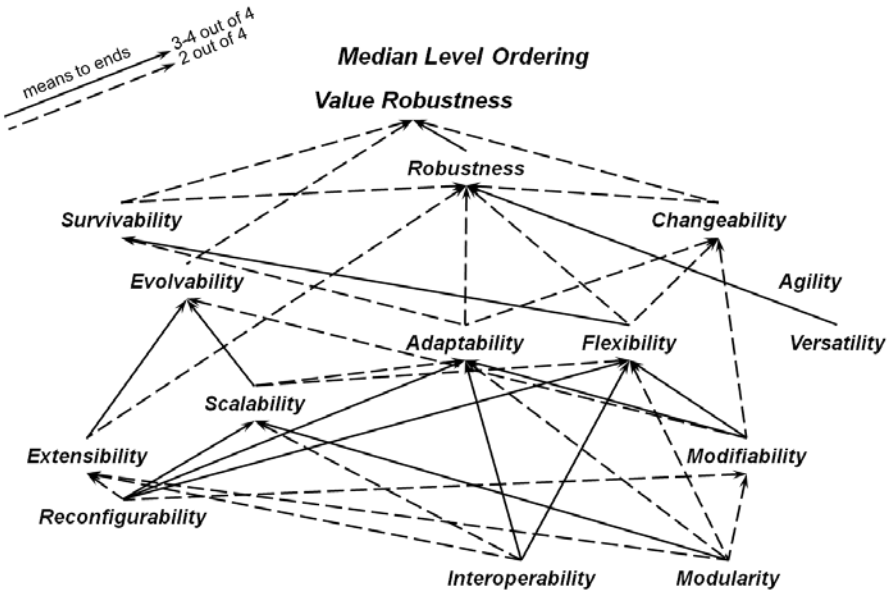


Figure 9. Median level ordering means-ends hierarchy from study

Feedback from the study indicated that the "bottom" ilities contained some different "sense" than the higher ilities. In particular, modularity and interoperability were viewed as a different perspective on a system lifecycle property than the others and implied particular architectural choices. This insight corresponds to the "architecture principle" concept asserted by Fricke and Schulz (2005) and implies that these ilities may belong to a different semantic field than the others.

Generalizing the Changeability Statement: A Semantic Basis

Building upon the insights from the various approaches for describing ilities above, this paper now describes an initial approach for creating a prescriptive semantic basis for

consistently representing ilities within a particular semantic field. At this time, the semantic basis, made up of fourteen categories, is believed to span the *change-type* ility semantic field and excludes the *architecture-type* semantic field that includes "bottom" ilities (de Weck, Ross, and Rhodes 2012) and "architecture principles" (Fricke and Schulz 2005) described above.

Beginning with the change agent and change effect as two categories for defining a change and the resulting applicable ilities, a larger set of categories are proposed for defining a larger set of possible changes for a system. The fourteen categories, which together form the semantic basis, are intended to collectively define a change in a system, thereby creating a consistent basis for specifying change-type ilities in formal statements. A system can be verified to display the quality described in the statement and therefore be traceable to a desired higher order system property.

The fourteen categories are: cause, context, phase, agent, impetus, impetus nature, impetus parameter, impetus destination state size, impetus aspect, outcome effect, outcome parameter, outcome destination state size, outcome aspect, level of abstraction, and value qualities of the change. Unique choices for each of these categories, when applied to a particular system parameter will formulate the change-type ility statement. The fourteen categories are illustrated in Figure 10.

The semantic basis aims to capture the essential differences among change-type ilities through specification of the following general change statement with regard to a particular system parameter:

In response to "cause" in "context", desire "agent" to make some "impetus parameter change" in "system" resulting in "outcome parameter change" that is "valuable."

Application of the semantic basis begins with a user generating a change statement. The change statement is refined and assigned categorical choices within the basis, with the intention that the applicable ilities will emerge from the specified change statement. In this way, a user does not need to use a particular ility label a priori, thereby avoiding the semantic ambiguity in the terms. If the basis accurately and completely describes the underlying categories for change-type changes, then a user should be able to describe any change-type ility through the basis.

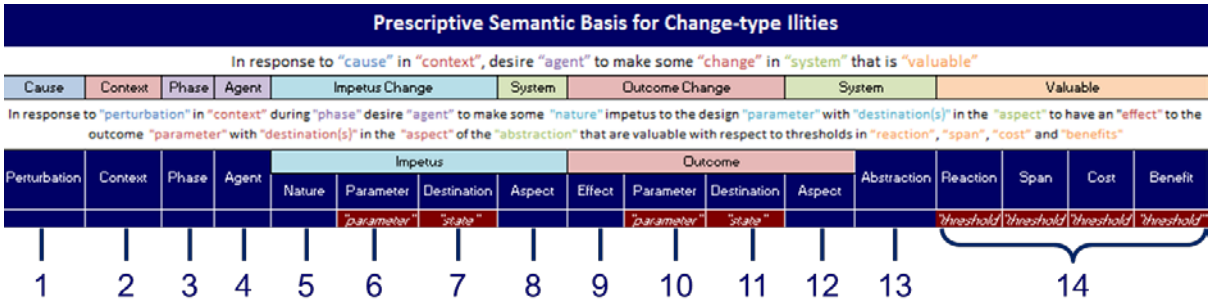


Figure 10. Change-type prescriptive semantic basis in 14 categories.

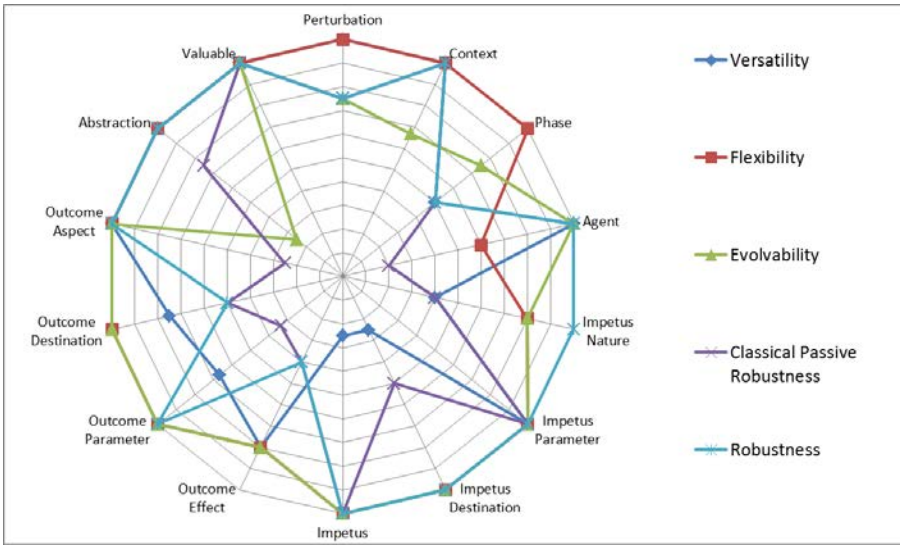


Figure 11. A radar plot depiction of change-type ilities in a 14 category basis

These fourteen categories can be visualized using a radar plot, with different ilities tracing a different shape (Figure 11).

In order to validate the proposed use of the basis, two activities need to be accomplished: refinement of the basis itself, both in terms of complete categories and in terms of choices within those categories. Additionally, if the use of particular ility terms is ultimately desired, a mapping between patterns in the basis and ility terms needs to be generated. If the latter is accomplished, then any usage of the basis for specifying particular change statements will result in consistently derived ility term labels. It is hypothesized that particular change-type ilities will correspond to particular choice(s) in this basis. In this way, a less ambiguous mechanism for specifying ilities can be created.

An example of using the semantic basis for mapping ility terms is illustrated in Figure 12. The results in the figure are presently a work in progress and are meant for illustration purposes only.

Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Impetus Change				System	Outcome Change				System	Valuable				
In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the design "parameter" with "destination(s)" in the "aspect" to have an "effect" to the outcome "parameter" with "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"																		
Perturbation	Context	Phase	Agent	Impetus					Outcome				Abstraction	Reaction	Span	Cost	Benefit	
				Nature	Parameter	Destination	Aspect		Effect	Parameter	Destination	Aspect						
					"parameter"	"state"				"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"	
disturbance	circumstantia	pre-ops	internal	increase	level	one	form		increase	level	one	form	architecture	sooner	shorter	less	more	
shift	general	ops	external	decrease	set	few	function		decrease	set	few	function	design	later	longer	more	less	
none	any	inter-LC	either	not-same	any	many	operations		not-same	any	many	operations	system	always	same	same	same	
any		any	none	same		any	any		same		any	any	any	any	any	any	any	
			any	any					any									

																		Ility Label
shift		ops							same	"Value"	few							Value Robustness
disturbance		ops							same	"Value"	few							Value Survivability
shift		ops							same		few							Robustness
shift		ops		not-same					same		few							Active Robustness
shift		ops		same		few			same		few							Passive Robustness
shift		ops	none	same		few			same	level	few	form	system					Classical Passive Robustness
disturbance		ops							same		few							Survivability
			either	not-same					not-same									Changeability
shift	general	inter-LC		not-same					not-same				architecture					Evolvability
			internal	not-same					not-same									Adaptability
			external	not-same					not-same									Flexibility
				not-same					not-same	level								Scalability
				not-same					not-same	set								Modifiability
		ops	either	not-same					increase	set								Extensibility
				not-same					not-same	any					shorter			Agility
				not-same					not-same	any				sooner				Reactivity
		ops		same	"Element set"	one	form	not-same	"Link set"			form						Form Reconfigurability
		ops		same	"Element set"	one	operations	not-same	"Order set"			operations						Operational Reconfigurability
		ops		same		one	form/ops	not-same	set		few/many							Versatility
		ops		same		one	form/ops	not-same	set		few/many	function						Functional Versatility
		ops		same		one	form/fnct	not-same	set		few/many	operations						Operational Versatility
		ops		same		one	fnct/ops	not-same	set		few/many	form						Substitutability

Figure 12. Using the semantic basis to consistently identify ility term labels

Using the basis

Beginning with a desired system parameter change statement, the user fills in the blanks, signified by the double quotation marks (“ ”) in the general form of the statement (illustrated in Figure 13 below). It is important that this statement be specific to a chosen system parameter, as currently the change-type ilities are conceived as being defined only in relation to particular system parameters (i.e. there is no such thing as “generalized flexibility” as one can always find exceptions). Working through the chart, a user fills in the categories with corresponding options for that particular change, which results in a categorized change description. For example, “in response to spike in fuel prices”, desire “customer” to “dramatically increase fuel efficiency of car during operations” that “costs less than $3000 and one week.” This statement requests “scalability” in “fuel efficiency” and specifies the value proposition (what is deemed “valuable” is that it cost less than $3000 and take less than a week to execute this change). Alternative cars can be evaluated on this basis, with those cars able to meet the value proposition displaying more valuable scalability in this regard. If ility labels have been previously mapped to the basis, then one can automatically derive the applicable ility terms implied by the categorized change statement.

A blank template for specifying change-type ilities using the basis is illustrated in Figure 13.

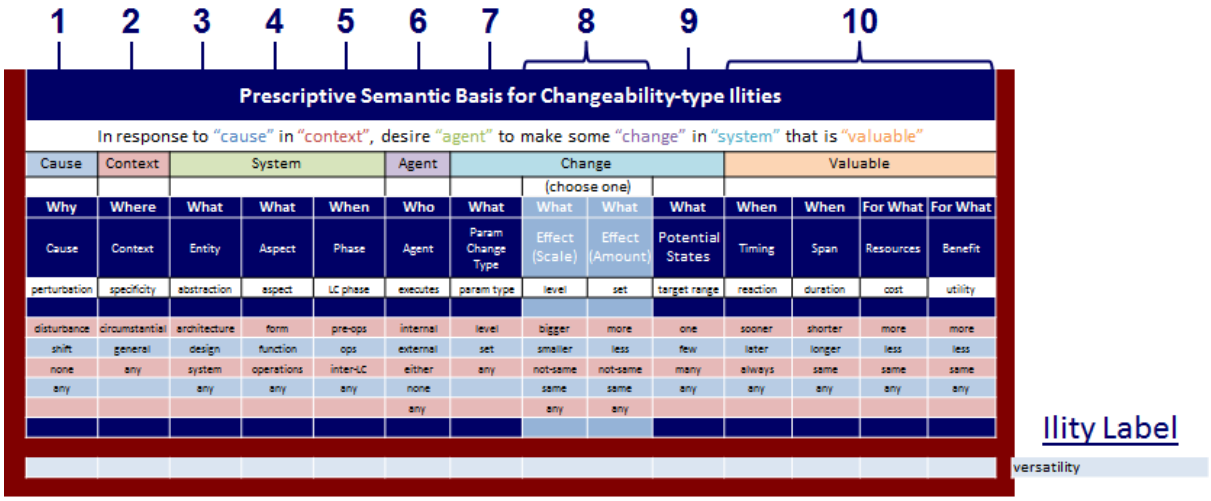


Figure 13. A template for constructing ility labels from 14 category basis

The semantic basis will now be described in more detail. Each item in double quotes (“ “) corresponds to choices that differentiate the change-type ilities. As a prerequisite to specifying a change statement, one must have in mind what “parameter” of the “system” is being affected. The “cause” column refers to whether this is in response to a perturbation that is finite duration or an epoch shift, or no perturbation at all (none), or it doesn’t matter (any). The “context” column refers to whether the response is desired in a particular epoch (circumstantial) or many epochs (general), or it doesn’t matter (any). The “system” is described by several columns. The “entity” is the level of abstraction of the system corresponding to the architecture level (encompasses many possible designs), design level (intended realization, possibly encompassing many possible actual realizations), system level (actual realization), or doesn’t matter (any). The “aspect” column refers to whether the response in the “system” is in terms of form, function, operations, or doesn’t matter (any). Sometimes the particular phase of lifecycle when the response occurs matters, with choices

here of pre-ops (during development, pre-operations), ops (during operations), inter-LC (between lifecycles), or doesn't matter (any). The "agent" is the force that sets the response into motion; active response requires a change agent, which can be internal or external to the "system" boundary, while passive response doesn't require an agent (none), or it may not matter (any). The desired "Δ" in the "system" is described by three columns: firstly, what type of parameter change (in level, set or any), secondly, the effect of the change (depending on whether it is a set or level, could get smaller, bigger, either, or same or doesn't matter=any), and thirdly, there could be a target number of end states ranging from one to few to many to any. Lastly, a distinct set of columns relate to whether the desired change response is actually valuable. This set of columns relates to time (timing and duration), resources (cost), and benefit (utility). This set is separated from the rest of the columns since it represents a coupled set of tradeoffs, while the other columns can be used to identify what is being specified and evaluated. For example, one might be willing to accept later, slower, and more expensive if it provides greater utility, but if it provides less utility, then maybe it should be sooner, faster, and cheaper. Multiple combinations of these columns could result in the same ility statement being valid.

Discussion

As research progresses on the prescriptive semantic basis, several open questions remain. For the particular basis described in this paper, what types of ilities can be represented in the basis? Can or should the basis be expanded or modified? What are the appropriate basis categories? What are appropriate choices within each category?

With regard to mapping specifications within the basis to particular ility term labels, are there consensus patterns in matching ilities to the basis given particular definitions for each ility? Are there consensus patterns for given ility terms without provided definitions (i.e. is there some inherent, general meaning within an ility term that can be more consistently expressed using the basis than through traditional English phrase-based definitions)?

More generally, this research begins to structure the question regarding what semantic fields span the general set of ilities. Preliminary results indicate that at least three semantic fields may exist in the general set of "ilities" including change-type, architecture-type, and new ability-type (the last type includes such ilities as "auditability," "learnability," and "drinkability" and is described in Ross (2008)). Identifying and classifying the current existing ility terms into appropriate semantic fields will serve to eliminate ambiguity in meaning, usage, and application, as well as allow for the explicit consideration of trade-offs within the semantic field. A consistent basis within a field can allow for direct comparison of its members; for example kinship terms clearly distinguish between the meaning of uncle and cousin, even though a single person could serve in both roles. Using the prescriptive semantic basis approach also allows one to consider whether each semantic field can be represented with an internally consistent basis.

Revisiting the concept of relationships amongst the ilities, the basis can provide a first order approximation to clarify semantic differences amongst ilities within a particular semantic field. For example the difference between "flexibility" and "adaptability" is whether the change agent is external to or internal to the system's boundary, respectively. The basis will also point out how a given change statement can display multiple ilities simultaneously. For example, agility is with regard to how quickly the change can be executed, so one could desire an agile, scalable change to describe a quick and level-increasing system parameter

change. In this way, the difficulty in placing "agility" within the means-ends hierarchy study, described above, can be clarified, as any change could get labeled as agile, depending on its application and desires of the change statement writer. An additional investigation will be needed in order to clarify the relationships between different semantic fields. For example, how are members of the "architecture-type" semantic field related to the "change-type" semantic field? Preliminary work investigating design/architecture principles have begun to describe relationships between particular ilities and principles, but a unifying understanding of the relationship between semantic fields is not yet mature[4].

Given a stable, validated basis, can practitioners or academics use the basis to generate change statements, which will automatically label with the appropriate ilities? Do the ility term labels resonate with the users?

The purpose of the research is not to generate more definitions, but rather, unambiguous, verifiable, standardized representations of desired system properties. One of the possible emergent results of this work may be the discovery of "new" ilities that do not yet have ility term labels, and yet may represent important desired system lifecycle properties, such as the distinction between functional versatility and operational versatility seen in Figure 14[5].

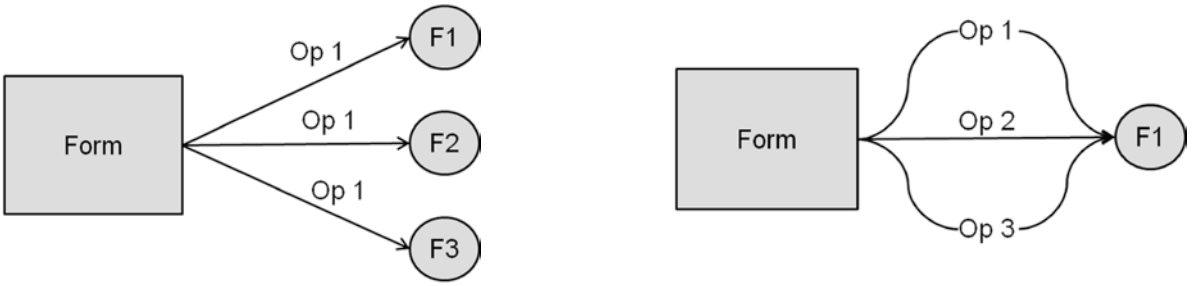


Figure 14. Functional and operational versatility as derived from using the semantic basis

The ultimate goal of this research is to develop the basis or bases to be prescriptive instrument(s) for spanning the semantic fields whose union encompasses all "ilities." With such an instrument, practitioners can have a consistent and (potentially) complete list of possible ilities to consider for their systems, as well as a means to create verifiable requirements and system specifications. Academics can have a consistent basis for enhancing ility-related research and a means to advance the quantification of and

[4] See for example Fricke and Schulz (2005) for changeability-related architecture principles; see Beesemyer et al. (2011) for evolvability-related design principles; see Banerjee and de Weck (2004) for flexibility-related design principles; see Siddiqi and de Weck (2008) for reconfigurability-related design principles; see Richards et al. (2009) for survivability-related system design principles; and see Mekdeci et al. (2011) for survivability-related SoS design principles.

[5] The distinction in specification between "functional versatility" and "operational versatility" is shown in Figure 12, with the former meaning that one could achieve multiple functions with similar form and operations, and the latter meaning that one could achieve similar function with multiple operations, but similar form. These different types of versatility would have different ramifications on design and operations decisions for a system.

clarification of relationships amongst ilities in general, as well as a means to educate future generations of engineering students so that one day ilities *can* become part of the lexicon of successful project managers.

References

Akmajian, A., Demers, R.A., Farmer, A.K., and Harnish, R.M. (2001), *Linguistics*. MIT Press: Cambridge, MA.

Banerjee, P., and de Weck, O.L. (2004), "Flexibility strategy— valuing flexible product options," INCOSE/ICSE Conference on Synergy between Systems Engineering and Project Management, Las Vegas, NV, 2004.

Beesemyer, J.C., Fulcoly, D.O., Ross, A.M., and Rhodes, D.H. (2011), "Developing Methods to Design for Evolvability: Research Approach and Preliminary Design Principles," *9th Conference on Systems Engineering Research*, Los Angeles, CA, April 2011.

Boehm, B., Huang, L., Jain, A., and Madachy, R., "The Nature of Information System Dependability: A Stakeholder/Value Approach," USC Report USC-CSE-2004-520, October 2004.

Cotton, L.D., Haase, G.A., Havlicek, J.D., and Thal, A.E. (2009), "Value-Driven Enterprise Architecture Score (VDEA-Score): A Means of DoDAF Architecture Evaluation," *7th Conference on Systems Engineering Research*, Loughborough, UK, April 2009.

de Neufville, R. and Scholtes, S. (2011), *Flexibility in Engineering Design*, MIT Press: Cambridge MA.

de Weck, O.L., Roos, D., and Magee C. (2011), "Life-cycle Properties of Engineering Systems," in *Engineering Systems*, MIT Press: Cambridge, MA.

de Weck, O.L., Ross, A.M., and Rhodes, D.H. (2012), "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)", *3rd International Engineering Systems Symposium*, CESUN 2012, TU Delft, 18-20 June 2012.

Engel, A., and Browning, T. (2008), "Designing for Adaptability by Means of Architecture Options," *Systems Engineering*, Vol. 11, No. 2.

ESD Symposium Committee (2001), "ESD Terms and Definitions," *MIT Engineering Systems Division Working Paper Series*, ESD-WP-2002-1, Version 12, October 2001. [Available online at: http://esd.mit.edu/WPS/].

Fricke, E. and Schulz, A.P. (2005), "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout their Entire Lifecycle," *Systems Engineering*, Vol. 8, No. 4, pp. 342-359.

ISO/IEC (2001), ISO/IEC 9126-1: "Software engineering — Product quality — Part 1: Quality model."

Mekdeci, B., Ross, A.M., Rhodes, D.H., and Hastings, D.E. (2011), "Examining Survivability of Systems of Systems," *INCOSE International Symposium*, Denver, CO, June 2011.

Nilchiani, R. (2005), "Measuring Space Systems Flexibility: A Comprehensive Six-element Framework," *PhD in Aeronautics and Astronautics*, MIT, Cambridge MA.

Richards, M.G., Ross, A.M., Hastings, D.E., and Rhodes, D.H. (2009), "Survivability Design Principles for Enhanced Concept Generation and Evaluation," *INCOSE International Symposium*, Singapore, July 2009.

Ross, A.M. (2006), "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration," *PhD in Engineering Systems*, MIT, Cambridge, MA.

Ross, A.M. (2008), "Defining and Using the New "ilities," *MIT SEAri Working Paper Series*, WP-2008-4-1, June 2008. [Available online at http://seari.mit.edu].

Ross, A.M., Rhodes, D.H., and Hastings, D.E. (2008), “Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value,” *Systems Engineering*, Vol. 11, No. 3, pp. 246-262.

Saleh, J. H., Mark, G.T., Jordan, N.C. (2009), "Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems," *Journal of Engineering Design*, Vol. 20, No. 3. pp. 307-323.

Siddiqi A., de Weck O.L. (2008), "Modeling Methods and Conceptual Design Principles for Reconfigurable Systems", *Journal of Mechanical Design*, 130, 101102, October 2008

Wikipedia (2012), “List of System Quality Attributes (redirect from ‘ilities’)”, http://en.wikipedia.org/wiki/Ilities (last accessed 28 Feb 2012).

Biography

Adam M. Ross is a Research Scientist in the MIT Engineering Systems Division (ESD) and the co-founder of the MIT Systems Engineering Advancement Research Initiative (SEAri). His research focuses on managing unarticulated value, designing for value robustness, and dynamic tradespace exploration for complex systems. Dr. Ross received his Ph.D. from MIT ESD and has published over 50 papers on changeability in system design and analysis. He has work experience with government, industry, and academia including NASA Goddard, JPL, the Smithsonian Astrophysical Observatory, Boeing Satellite Systems, MIT, Harvard, and Florida State Universities, performing both science and engineering research.

J. Clark Beesemyer is a research assistant in SEAri and a masters student in the MIT Department of Aeronautics and Astronautics. He received a B.S. in Astronautical Engineering from the United States Air Force Academy. There he served as the chief engineer for FalconSAT-5, a small satellite program exploring the effects of space weather. He now serves as a 2nd Lieutenant in the U.S. Air Force and will be attending pilot training at the conclusion of his masters program.

Donna H. Rhodes is co-founder and director of SEAri and a Senior Lecturer in Engineering Systems at MIT, where she is also a Principal Research Scientist. Her research interests include predictive architecting of socio-technical systems and anticipatory capacity in enterprises. Dr. Rhodes has 20 years of experience in the aerospace, defense systems, systems integration, and commercial product industries. Prior to joining MIT, she held senior level management positions at IBM Federal Systems, Lockheed Martin, and Lucent Technologies in the areas of systems engineering and enterprise transformation. Dr. Rhodes is a past president and Fellow of INCOSE, and the 2005 recipient of the INCOSE Founders Award.

Appendix: Description of Categories for Semantic Basis

Using the semantic basis for identifying implied ility labels begins with a user generating a change statement. This statement will be parsed according to the categories of the basis to determine applicable ility label(s) that can be used to describe the change. The structure of the change statement is generalized below: [NEEDS UPDATE TO 14-D]

In response to "perturbation" in "context", desire "agent" to make some "effect" to the "system parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits".

Application of this statement to a specific change requires the " " marks in the general form of the statement to be replaced with applicable terms. It is important that this statement is phrased relative to specific impetus and outcome parameters.

Working through the change-type semantic prescriptive basis

For the purposes of working through the categories of the basis, the following description will give examples that may apply to a passenger aircraft. These tables may be used to clarify the questions to be used when examining different ility statements.

Perturbation

disturbance

shift

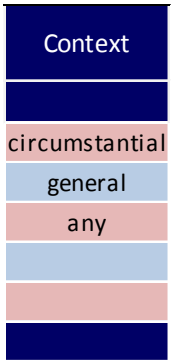
none

any

1. Perturbation

This section refers to the cause or perturbation to the system. In a sense, it answers "why?" - or what the system is responding to.

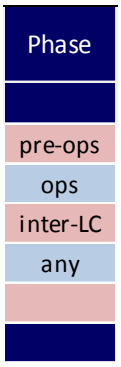
Disturbance	The perturbation is some finite (short) duration of time, with a defined start and end time (e.g. a spike in temperature or a gust of wind) and relatively short in timescale compared to the system timescale
Shift	The perturbation is imposed with no end in the foreseeable future. The perturbation is constant and makes up part of the new environment in which the system must operate (e.g. a new federal regulation or environment of operation)
None	There is no perturbation. This response is for the system changes that do not occur in response to some perturbation
Any	A broader perturbation that may be either finite or a shift in environment. This response captures both of the types of perturbations)



2. Context

This section refers to the context on which the system responds to the perturbation

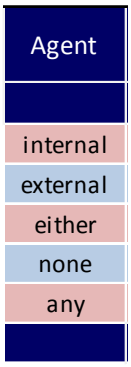
Circumstantial	The desired response is applicable only in certain conditions or a specific epoch (e.g. mitigating radio interference in critical phases of flight, such as take-off and landing)
General	The desired response is applicable to multiple conditions or many epochs (e.g. increase energy awareness desiring airliners to be more efficient)
Any	The context doesn't matter and could be either circumstantial or general



3. Phase

This section refers to the lifecycle phase in which the change occurs or "when?"

Pre-Ops	The change occurs in the design process before the system makes it to operational stages (e.g. a change in the concept or any change during initial design, testing, and integration of an aircraft)
Ops	The change occurs in the operational stages of the system (e.g. the redesign to the Airbus 380 engines)
Inter-LC	The change occurs between lifecycles (e.g. taking the design of an aircraft and altering it to create a new aircraft)
Any	The change could occur in any phase of the system lifecycle



4. Agent

This section refers to the change agent responsible for the change or "who?"

Internal	The change agent is internal to defined system boundaries (e.g. the autopilot recognizing a need to change course)
External	The change agent is external to defined system boundaries (e.g. an aircraft engineer upgrading a component such as landing gear)
Either	The change agent is either internal or external, the location of the change agent does not matter
None	There is no change agent involved, a passive response (e.g. the aircraft being robust to random gusts of wind in flight)

Any	The change agent could be external or internal to the system or none

Effect
increase
decrease
not-same
same
any

5. Effect

This section refers to the effect or directionality of the change to the *parameter*

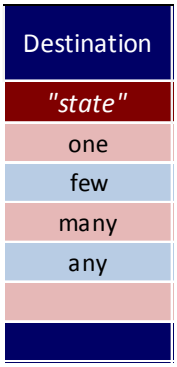
Increase	The change is an increase in parameter (e.g. an increase in cargo capacity or types of cargo that may be carried)
Decrease	The change is a decrease in parameter (e.g. a decrease in rivets needed or a decrease in amenities offered like meals, snacks, or movies)
Not-Same	The change in parameter just has to be different (e.g. a change in the exterior of the aircraft)
Same	There is no change in parameter (e.g. maintaining a level or set parameter, like lift, across a perturbation for robustness or survivability)
Any	The change in parameter doesn't matter and could be any of the above

Parameter (Type)
"parameter"
level
set
any

6. Parameter type

This section refers to the type of parameter and includes a reference box for the specific parameter in question. This parameter is essential for the change statement to be clear and can be an attribute or a system parameter (lift, value, level of thrust, set of crew)

Level	The change is in the scale of the parameter (e.g. an increase in fuel capacity to the aircraft)
Set	The change is in the amount of parameters (e.g. changing the different types of runways the aircraft can land on)
Any	The change could be to either parameter change type



7. Destination

This section refers to the target range of potential states of the parameter as a result of the change and includes a reference box if a specific range or state is necessary (increase the level of thrust to “X pounds of force”

One	There is only one desired end state of the parameter due to the change (e.g. the landing gear being extended for landing)
Few	There are a few (countable number) different desired states due to the change (e.g. altering the in-flight media for varying length in travel times)
Many	There are many (uncountable) different end states due to the change (e.g. thrust level attainable through the lever (continuous))
Any	There is no specific number of target end states

8. Aspect

This section refers to the aspect of the abstraction being changed

Form	The change is in the physical components of the system (e.g. a change in the engine or wing design of an aircraft)
Function	The change is in the purpose of a system or system component (e.g. Using the in-flight media system to deliver safety and flight information)
Operations	The change is in how the system is used or implemented (e.g. a new set of flight rules to maximize fuel efficiency)
Any	The change could be to any of these aspects of the system

Abstraction

architecture

design

system

any

9. Abstraction

This section refers to the level of abstraction of the system being affected or what entity of the system

Architecture	The change is in the architecture of the system, where architecture is the higher level of mapping form to function of the system, affecting multiple possible designs (e.g. a change in the architecture of the Boeing 747)
Design	The change is in the design or the intended realization of a system (e.g. the design for the 747-8 intercontinental or other versions of the 747)
Realized System	The change is in the realized system. The particular system has been built, and the change is applied to this realized system (e.g. a specific aircraft with tail number ##)
Any	The change could be to any of these levels of abstraction of the system

** Another helpful way to think about levels of abstraction is the Apple iPod. The architecture would be the iPod in general, the design would be the more specific model such as the 16GB or 32GB iPod, and the system would be one specific instance of that design. This construct is degenerative for unique systems where the architecture, design, and instance may be represented by the same entity, like the International Space Station (ISS). More information on this may be found in Chapter 3 in the evolvability section.

The following columns refer to the value of the change described

Note: the "valuable" section does not necessarily have to be specified, and multiple specifications of these columns could be deemed equally "valuable" (e.g. less benefit may be okay if it happens sooner, or slower may be okay if it is cheaper or has more benefit). These statements are in comparison to some "baseline" expectation on change "cost," often implicit to status quo change mechanisms (e.g. current product development schedules and budgets) or to a specified threshold amount.

Reaction
"threshold"
sooner
later
always
any

10. Reaction

This section refers to the change start time or when the change initiates and has a threshold value that may be used as a reference (a specific target value, or reference to current value of the unchanged system)

Sooner	The change initiates earlier than baseline/threshold (e.g. the autopilot recognizing a need to change course earlier than a previous model)
Later	The change initiates later than baseline/threshold (e.g. the autopilot recognizing a need to change course later than a previous model)
Always	The change or lack of change is always present and therefore does not initiate (e.g. the aircraft being robust to random gusts of wind in flight)
Any	It doesn't matter when the change initiates

Span
"threshold"
shorter
longer
same
any

11. Span

This section refers to the time it takes to change with respect to a baseline or comparable system or threshold value

Shorter	The change duration is shorter than the baseline/threshold (e.g. the autopilot changes course in less time than a previous model)
Longer	The change duration is longer than the baseline/threshold (e.g. the landing gear takes longer to switch positions than a previous model)
Same	There is no change in duration with respect to the baseline/threshold
Any	The change could be any duration

Cost
"threshold"
less
more
same
any

12. Cost

This section refers to the resources required for the change with respect to a baseline or comparable system or threshold value

Less	The change costs less than it would in the baseline/threshold (e.g. the change in course or altitude uses less fuel than the previous model)
More	The change costs more than it would in the baseline/threshold (e.g. re-painting a bigger plane will cost more than a smaller plane)
Same	There change costs the same as it would in the baseline/threshold
Any	The change agent could be external or internal to the system

Benefit

"threshold"

more

less

same

any

13. Benefit

This section refers to the utility as a result of the change with respect to the baseline system or threshold value

More	The change results in an more utility than the baseline/threshold (e.g. better brakes allowing shorter stopping times)
Less	The change results in an less utility than the baseline/threshold (e.g. a wing design the creates less lift than the baseline)
Same	The change results in the same utility as the baseline/threshold
Any	The change agent could be external or internal to the system